CS-E4710 Machine Learning: Supervised Methods

Lecture 2: Statistical learning theory

Juho Rousu

September 15, 2020

Department of Computer Science
Aalto University

Generalization

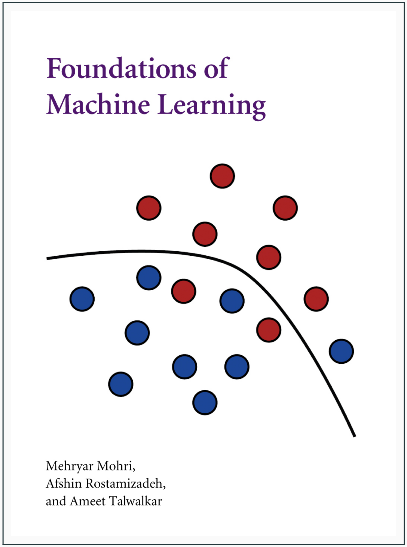
- Our aim is to predict as well as possible the outputs of future examples, not only for training sample
- We would like to minimize the **generalization error**, or the (true) **risk**

  $$R(h) = \mathbb{E}_{(\mathbf{x},y)\sim D}\left[\, L(h(\mathbf{x}), y)\,\right],$$

  where $L(y, y')$ is a suitable loss function (e.g. zero-one loss)
- Assuming future examples are independently drawn from the same distribution D that generated the training examples (i.i.d assumption)
- But we do not know D!
- What can we say about $R(h)$ based on training examples and the hypothesis class $\mathcal{H}$ alone? Two possibilities:
  - Empirical evaluation through testing
  - **Statistical learning theory (Lectures 2 and 3)**

Additional reading

- This lecture mostly follows Mohri et al: chapter 2
- The book goes deeper in the theory (e.g. proofs of theorems) than what we do in the course



Foundations of Machine Learning

Mehryar Mohri,
Afshin Rostamizadeh,
and Ameet Talwalkar

Probably approximately correct learning

Probably Approximate Correct Learning framework

- Probably Approximate Correct (PAC) Learning framework formalizes the notion of generalization in machine learning
- Ingredients:
    - input space X containing all possible inputs x
    - set of possible labels $\mathcal{Y}$ (in binary classification $\mathcal{Y} = \{0, 1\}$)
    - Concept class $\mathcal{C}$ containing concepts $C : X \mapsto \mathcal{Y}$ (to be learned), concept C gives a label $C(x)$ for each input x
    - unknown probability distribution D
    - training sample $S = (x_1, C(x_1)), \ldots, (x_m, C(x_m))$ drawn independently from D
    - hypothesis class $\mathcal{H}$, in the basic case $\mathcal{H} = \mathcal{C}$ but this assumption can be relaxed
- The goal in PAC learning is to learn a hypothesis with a low generalization error

$$R(h) = \mathbb{E}_{x \sim D}\left[\, L_{0/1}(h) \,\right] = \Pr_{x \sim D}(h(x) \neq C(x))$$

PAC learnability

- A class $\mathcal{C}$ is **PAC-learnable**, if there exist an algorithm $\mathcal{A}$ that given a training sample S outputs a hypothesis $h_S \in \mathcal{H}$ that has generalization error satisfying

$$Pr(R(h_S) \leq \epsilon) \geq 1 - \delta$$

  - for **any** distribution D, for arbitrary $\epsilon, \delta > 0$ and sample size $m = |S|$ that grows at polynomially in $1/\epsilon$,$1/\delta$
  - for **any** concept $C \in \mathcal{C}$
- In addition, if $\mathcal{A}$ runs in time polynomial in m,$1/\epsilon$,and $1/\delta$ the class is called **efficiently PAC learnable**

Interpretation

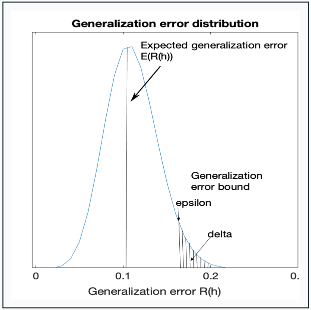
Let us interpret the bound

$$Pr(R(h_S) \leq \epsilon) \geq 1 - \delta$$

- ϵ sets the level of generalization error that is of interest to us, say we are content with predicting incorrecly 10% of the new data points: $\epsilon = 0.1$
- $1 - \delta$ sets a level of confidence, if we are content of the training algorithm to fail 5% of the time to provide a good hypothesis: $\delta = 0.05$
- We want the requirement for training data and running time grow modestly when we make ϵ and δ stricter: requirement of polynomial growth
- The event "low generalization error", $\{R(h_S) \leq \epsilon\}$ is considered as a random variable because we cannot know beforehand which hypothesis $h_S \in \mathcal{H}$ will be selected by the algorithm

Generalization error bound vs. test error

- Generalization error bounds concern the tail of the error distribution
    - We wish a high generalization error to be a rare event
- Expected generalization error might be considerably lower
    - Analyzing average behaviour where most distributions and concepts are "not bad"

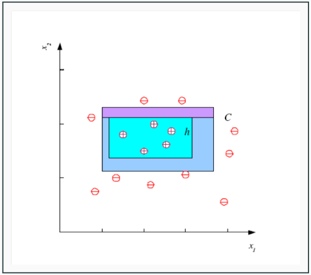


Example: learning axis-parallel rectangles

PAC learning the "family car"

Assumptions

- True concept C ("family car") can be represented with a axis-parallel rectangle
- Our algorithm chooses the smallest rectangle h_S that includes all positive training examples (the most specific hypothesis)
- The smallest rectangle is consistent, i.e. does not contain any negative examples

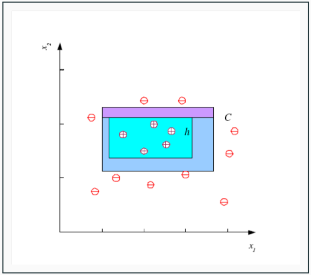


How many examples do we need to guarantee $Pr(R(h_S) \leq \epsilon) \geq 1 - \delta$?

PAC learning the "family car"

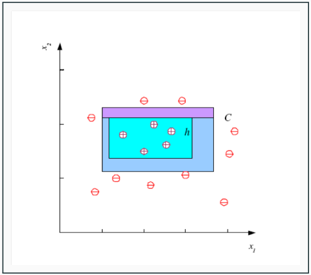
How many examples do we need to guarantee $Pr(R(h_S) \leq \epsilon) \geq 1 - \delta$

- The generalization error $R(h_S) = Pr(C \Delta h_S)$ is the measure of the symmetric difference $C \Delta h_S = \{x \in X | h_S(x) \neq C(x)\}$
- We need to bound the probability mass in the difference: $Pr(C \Delta h_S) < \epsilon$ given the knowledge that no randomly drawn example fell inside the region
- Draw 4 strips of probability mass $\epsilon/4$ (top, bottom, right, left) inside C; their union has probability mass $< \epsilon$



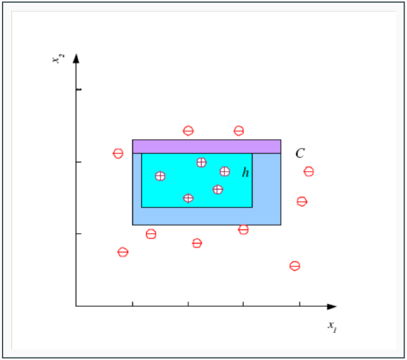
PAC learning the "family car"

- Events
  $A = \{h_S \text{ intersects all four strips}\}$,
  $B = \{R(h_S) < \epsilon\}$, satisfy $A \subseteq B$
- Complement events
  $A_C = \{h_S \text{ misses at least one strip }\}$,
  $B_C = \{R(h_S) \geq \epsilon\}$ satisfy $B_C \subseteq A_C$
- B_C is the bad event (high generalization error), we want it to have low probability
- In probability space, we have $Pr(B_C) \leq Pr(A_C)$
- Let us now upper bound $Pr(A_C)$



PAC learning the "family car"

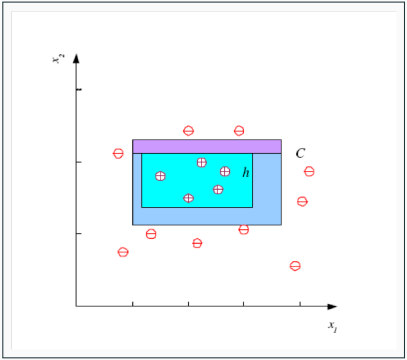
$$
\begin{aligned}
A_C &= \{h_S \text{ misses at least one strip }\} \\
&= \{h_S \text{ misses the left strip }\} \cup \\
&\quad \{h_S \text{ misses the right strip }\} \cup \\
&\quad \{h_S \text{ misses the top strip }\} \cup \\
&\quad \{h_S \text{ misses the bottom strip }\}
\end{aligned}
$$



PAC learning the ”family car”

- Each strip has probability mass $\epsilon/4$ by our design
- Probability of one example missing one strip: $1 - \epsilon/4$
- Probability of m examples missing one strip: $(1 - \epsilon/4)^m$ (m times repeated trial with replacement)
- Probability of all examples missing at least one of the strips:

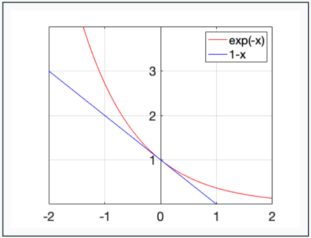
$$Pr(A_C) \leq 4(1 - \epsilon/4)^m$$



PAC learning the ”family car”

- We can use a general inequality $\forall x : (1 - x) < \exp(-x)$ to obtain:

$$Pr(R(h) \geq \epsilon) \leq 4(1-\epsilon/4)^m \leq 4 \exp(-m\epsilon/4)$$



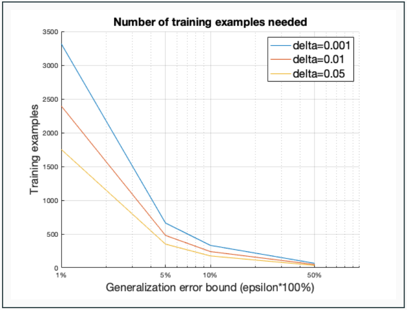
- We want this probability to be small $(< \delta)$:

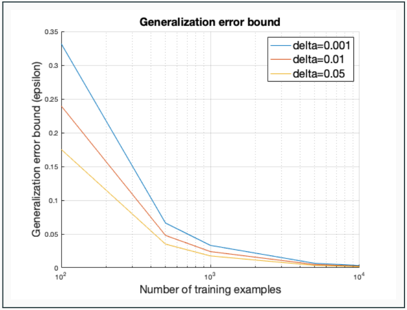
$$4 \exp(-m\epsilon/4) < \delta$$
$$\Leftrightarrow m \geq 4/\epsilon \log 4/\delta$$

- The last inequality is our first generalization error bound, a **sample complexity** bound to be exact

Plotting the behaviour of bound

- Left, the sample complexity, the number of examples needed to reach a given generalization error level is shown $m(\epsilon, \delta) = 4/\epsilon \log 4/\delta$
- Right, the generalization bound is plotted as a function of training sample size $\epsilon(m, \delta) = 4/m \log 4/\delta$
- Three different confidence levels (δ) are plotted

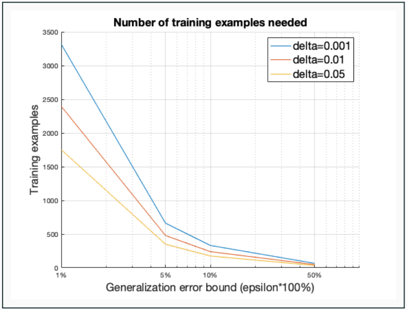


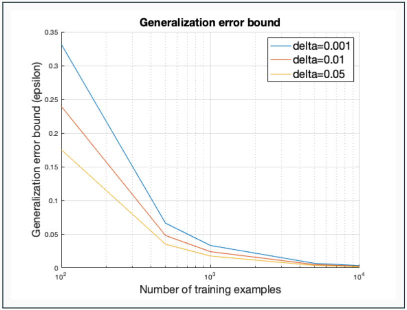


Plotting the behaviour of the bound

Typical behaviour of ML learning algorithms is revealed:

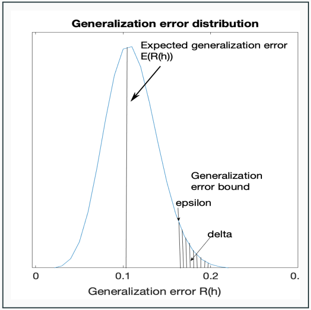
- increase of sample size decreases generalization error
- extra data gives less and less additional benefit as the sample size grows (law of diminishing returns)
- requiring high level of confidence (small δ) for obtaining low error requires more data for the same level of error





Generalization error bound vs. expected test error

- The error bounds hold for any concept from the class (e.g. "all vehicles" vs. "family car")
  - including difficult concepts, e.g. "Crossover SUV"
- They hold for **any** distribution D generating the data
  - Including adversially generated distributions (aiming to make learning harder)
- For these reasons empirically estimated test errors might be considerably lower than the bounds suggest



Notes

- The proof was very specific for the chosen class (axis-parallel rectangles), and not easy to immediately apply to other class
- In the following we show a general result for finite hypothesis sets
- Later analyze infinite hypothesis classes (Lecture 3)

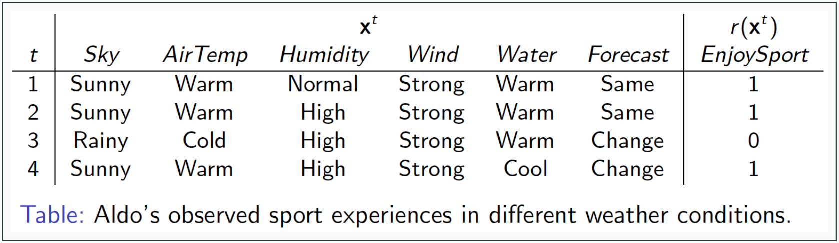
Guarantees for finite hypothesis sets

Finite hypothesis classes

- Finite concept classes arise when:
  - Input variables have finite domains or they are converted to such in preprocessing (e.g. discretizing real values), and
  - The representations of the hypotheses have finite size (e.g. the number of times a single variable can appear)
  - Subclasses of Boolean formulae, that expressions binary input variables (literals) combined with logical operators (AND, OR, NOT,...)
- Finite concept classes have been thoroughly analyzed hypothesis classes in statistical learning theory

Example: Boolean conjunctions

- Aldo likes to do sport only when the weather is suitable
- Also has given examples of suitable and not suitable weather
- Let us build a classifier for Aldo to decide whether to do sports today
- As the classifier we use rules in the form of boolean conjunctions (boolean formulae containing AND, and NOT, but not OR operators): e.g. if (Sky=Sunny) AND NOT(Wind=Strong) then (EnjoySport=1)



	$\mathbf{x}^t$						$r(\mathbf{x}^t)$
t	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	1
2	Sunny	Warm	High	Strong	Warm	Same	1
3	Rainy	Cold	High	Strong	Warm	Change	0
4	Sunny	Warm	High	Strong	Cool	Change	1

Table: Aldo's observed sport experiences in different weather conditions.

Finite hypothesis class - consistent case

- Sample complexity bound relying on the size of the hypothesis class (Mohri et al, 2012): $Pr(R(h_s) \leq \epsilon) \geq 1 - \delta$ if

$$m \geq \frac{1}{\epsilon}(\log(|\mathcal{H}|) + \log(\frac{1}{\delta}))$$

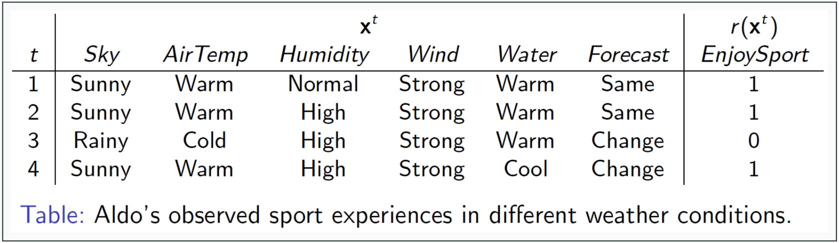
- An equivalent generalization error bound:

$$R(h) \leq \frac{1}{m}(\log(|\mathcal{H}|) + \log(\frac{1}{\delta}))$$

- Holds for any finite hypothesis class assuming there is a consistent hypothesis, one with zero empirical risk
- Extra term compared to the "family car" example is the term $\frac{1}{\epsilon}(\log(|\mathcal{H}|))$
- The more hypotheses there are in $\mathcal{H}$, the more training examples are needed

Example: Boolean conjunctions

- How many different conjunctions can be built ($=|\mathcal{H}|$)
- Each variable can appear with or without "NOT" or can be excluded from the rule = 3 possibilities
- The total number of hypotheses is thus 3^d, where d is the number of variables
- We have six variables in total, giving us $|\mathcal{H}| = 3^6 = 729$ different hypotheses

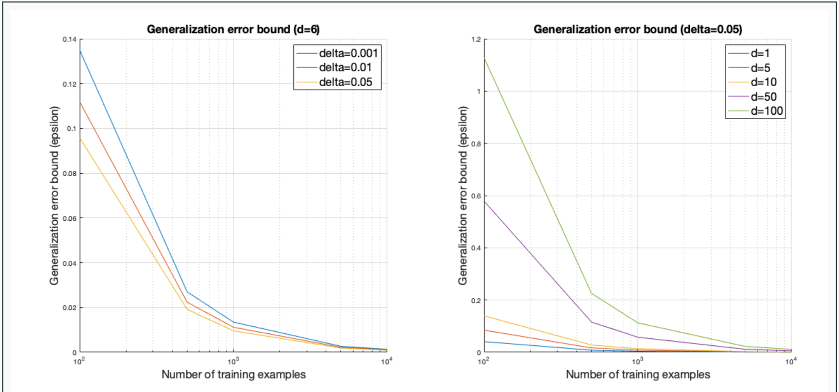


	$\mathbf{x}^t$						$r(\mathbf{x}^t)$
t	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	1
2	Sunny	Warm	High	Strong	Warm	Same	1
3	Rainy	Cold	High	Strong	Warm	Change	0
4	Sunny	Warm	High	Strong	Cool	Change	1

Table: Aldo's observed sport experiences in different weather conditions.

Plotting the bound for Aldo's problem using boolean conjunctions

- On the left, the generalization bound is shown for different values of δ, using $d = 6$ variables
- On the right, the bound is shown for increasing number of input variables d, using $\delta = 0.05$



Arbitrary boolean formulae

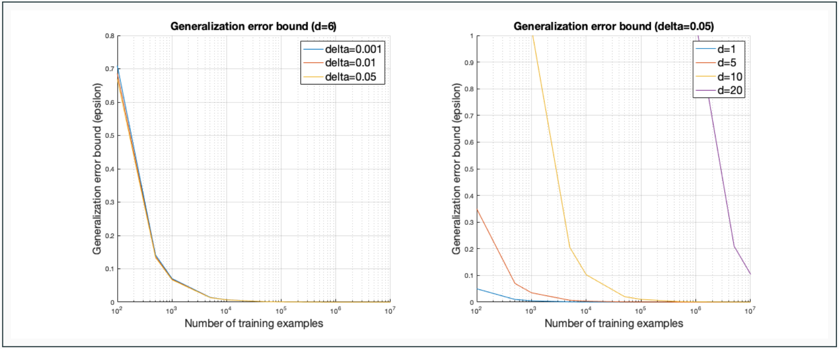
- What about using arbitrary boolean formulae?
- How many boolean formulae of d variables there are?
- There are 2^d possible input vectors, size of the input space is $|X| = 2^d$
- We can define a boolean formula that outputs 1 for an arbitrary subset of $S \subset X$ and zero outside that subset: $f_S(\mathbf{x}) = (\mathbf{x} = \mathbf{x}_1) OR (\mathbf{x} = \mathbf{x}_2) OR \cdots OR (\mathbf{x} = \mathbf{x}_{|S|})$
- We can pick the subset in $2^{|X|}$ ways (Why?)
- Thus we have $|\mathcal{H}| = 2^{2^d}$ different boolean formula
- Our generalization bound gives

$$m \geq \frac{1}{\epsilon}(2^d \log 2 + \log(\frac{1}{\delta}))$$

- Thus we need exponential number of examples with respect to the number of variables; the hypothesis class is considered not PAC-learnable!

Plotting the bound for Arbitrary boolean formulae

- With $d = 6$ variables we need ca. 500 examples to get bound below 0.07 (left picture)
- Increase of number of variables quickly raises the sample complexity to 10^6 and beyond (right picture)



Finite hypothesis class - inconsistent case

- So far we have assumed that there is a consistent hypothesis $h \in \mathcal{H}$, one that achieves zero empirical risk on training sample
- In practise this is often not the case
- However as long as the empirical risk $\hat{R}(h)$ is small, a low generalization error can still be achieved
- Generalization error bound (Mohri, et al. 2012): Let $\mathcal{H}$ be a finite hypothesis set. Then for any $\delta > 0$ with probability at least $1 - \delta$ we have for all $h \in \mathcal{H}$:

$$R(h) \leq \hat{R}(h) + \sqrt{\frac{\log(|\mathcal{H}|) + \log(2/\delta)}{2m}}$$

- We see the dependency from $\log |\mathcal{H}|/m$ as in the consistent case but now under square root
    - Slower convergence w.r.t number of examples

Stochastic scenario

Stochastic scenario

- The analysis so far assumed that the labels are deterministic functions of the input
- Stochastic scenario relaxes this assumption by assuming the output is a probabilistic function of the input
- The input and output is generated by a joint probability distribution D or $X \times \mathcal{Y}$.
- This setup covers different cases when the same input x can have different labels y
- Agnostic PAC learning studies the generalization guarantees in the stochastic scenario (will not be covered in this course)

Sources of stochasticity

The stochastic dependency between input and output can arise from various sources

- Imprecision in recording the input data (e.g. measurement error), shifting our examples
- Errors in the labeling of the training data (e.g. human annotation errors), flipping the labels some examples
- There may be additional variables that affect the labels that are not part of our input data

All of these sources could be characterized as adding noise (or hiding signal)

Bayes error and noise

- In the deterministic scenario, there is a target concept f that has zero generalization error $R(f) = 0$
- In the stochastic scenario, there is a minimal non-zero error for any hypothesis, called the **Bayes error**
- Bayes error is the minimum achievable error, given a distribution D over $X \times \mathcal{Y}$, by measurable functions $h : X \mapsto \mathcal{Y}$

$$R^* = \inf_{\{h | h \text{ measurable }\}} R(h)$$

- A hypothesis with $R(h) = R^*$ is called the **Bayes classifier**

Bayes error and noise

- The Bayes classifier can be defined in terms of conditional probabilities as

$$h_{Bayes}(x) = \mathrm{argmax}_{y \in \{0,1\}} Pr(y|x)$$

- The average error made by the Bayes classifer at $x \in X$ is called the **noise**

$$noise(x) = \min(Pr(1|x), Pr(0|x))$$

- Its expectation $E(noise(x)) = R^*$ is the Bayes error
- Remember that since we do not know D, we cannot actually compute the Bayes classifier!
    - It serves as a theoretical model of the best possible performance

Decomposing the generalization error

The generalization error of a hypothesis can be decomposed as follows

$$R(h) = R^* + \epsilon_{estimation} + \epsilon_{approximation}$$

- R^* is the Bayes error or noise, which depends on the task and cannot be avoided
- $\epsilon_{estimation} = R(h) - R(h^*)$ is the excess generalization error h has over the optimal hypothesis $h^* = \mathrm{argmin}_{h' \in \mathcal{H}} R(h')$ in the hypothesis class $\mathcal{H}$
- $\epsilon_{approximation} = R(h^*) - R^*$ is the approximation error due to selecting the hypothesis class $\mathcal{H}$ instead of the best possible hypothesis class (which is generally unknown to us)
- Note: The approximation error is sometimes called the **bias** and the estimation error the **variance**, and the decomposition **bias-variance decomposition**

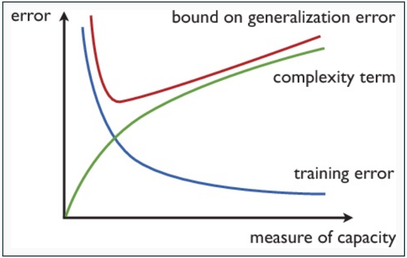
The trade-off between empirical error and complexity

The generalization error bounds we derived have the form

$$R(h) \leq \hat{R}(h) + O(\frac{\log |\mathcal{H}|}{m})$$

The second term can be interpreted as measuring the model complexity through the number of hypotheses in the class

- We have a trade-off: increasing model complexity or capacity generally decreases empirical error but increases the complexity term
- To minimize the generalization error, we should find a balance between the two terms



Controlling complexity

Two general approaches to control the complexity

- Selecting a hypothesis class, e.g. the maximum degree of polynomial to fit the regression model - this would typically be done prior to learning
- Regularization: penalizing the use of too many parameters, e.g. by bounding the norm of the weights (used in SVMs and neural networks) - this would typically happen automatically during learning (after setting the amount of regularization prior learning)

Measuring complexity

What is a good measure of complexity of a hypothesis class?

- Number of distinct hypotheses $|\mathcal{H}|$: works for finite $\mathcal{H}$ (e.g. models build form binary data), but not for infinite classes (e.g. geometric hypotheses such as polygons, hyperplanes, ellipsoids)
- Vapnik-Chervonenkis dimension (VCdim): the maximum number of examples that can be classified in all possible ways by choosing different hypotheses $h \in \mathcal{H}$
- Rademacher complexity: measures the capability to classify after randomizing the labels

Next lecture will focus on the two latter measures of complexity

Extra material: Proof outline of the PAC bound for finite hypothesis classes*

Proof outline* (Mohri et al., 2012)

- Consider any hypothesis $h \in \mathcal{H}$ with $R(h) > \epsilon$
- For h to be consistent $\hat{R}(h) = 0$, all training examples need to miss the region where h is making an error.
- The probability of this event is

$$Pr(\hat{R}(h) = 0 | R(h) > \epsilon) \leq (1 - \epsilon)^m$$

- m times repeated trial with success probability ϵ
- This is the probability that one consistent hypothesis has high error

Proof outline*

- But we do not need which consistent hypothesis h is selected by our learning algorithm
- Hence our result will need to hold for all consistent hypotheses
  - This is an example of **uniform convergence** bound
- We wish to upper bound the probability that some $h \in \mathcal{H}$ is consistent $\hat{R}(h) = 0$ and has generalization error $R(h) > \epsilon$ for a fixed $\epsilon > 0$:

$$Pr(\exists h \in \mathcal{H} | \hat{R}(h) = 0 \wedge R(h) > \epsilon)$$

- Above $\wedge$ is the logical "and"

Proof outline*

- We can replace $\exists$ by enumerating all hypotheses in $\mathcal{H}$ using logical-or ($\vee$)

$$
\begin{aligned}
&Pr(\exists h \in \mathcal{H} | \hat{R}(h) = 0 \wedge R(h) > \epsilon) = \\
&\qquad Pr(\hat{R}(h_1) = 0 \wedge R(h_1) > \epsilon) \vee Pr(\hat{R}(h_2) = 0 \wedge R(h_2) > \epsilon) \vee \cdots
\end{aligned}
$$

- Using the the fact that $Pr(A) \cup Pr(B) \leq Pr(A) + Pr(B)$ and $Pr(A \cap C) \leq Pr(A|C)$ for any events A,B and C the above is upper bounded by

$$
\begin{aligned}
\leq \sum_{h \in \mathcal{H}} Pr(\hat{R}(h) = 0 \wedge R(h) > \epsilon) \leq \sum_{h \in \mathcal{H}} Pr(\hat{R}(h) = 0 | R(h) > \epsilon) \\
\leq |\mathcal{H}|(1-\epsilon)^m
\end{aligned}
$$

- Last inequality follows from using the $Pr(\hat{R}(h) = 0 | R(h_1) > \epsilon) \leq (1-\epsilon)^m$ for the $|\mathcal{H}|$ summands

Proof outline*

- We have established

$$Pr(\exists h \in \mathcal{H} | \hat{R}(h) = 0 \wedge R(h) > \epsilon) \leq |\mathcal{H}|(1-\epsilon)^m \leq |\mathcal{H}| \exp(-m\epsilon)$$

- Set the right-hand side equal to δ and solve for m to obtain the bound:

$$\begin{aligned} \delta &= |\mathcal{H}| \exp\left(-m\epsilon\right) \\ \log \delta &= \log |\mathcal{H}| - m\epsilon \\ m &= \frac{1}{\epsilon}(\log(|\mathcal{H}|) + \log(1/\delta)) \end{aligned}$$